

# A Weighted Minimum Distance Using Hybridization of Particle Swarm Optimization and Bacterial Foraging

Muhammad Marwan Muhammad Fuad

Forskningsparken 3, Institutt for kjemi, NorStruct  
The University of Tromsø - The Arctic University of Norway  
NO-9037 Tromsø, Norway  
[marwan.fuad@uit.no](mailto:marwan.fuad@uit.no)

**Abstract:** In a previous work we used a popular bio-inspired algorithm; particle swarm optimization (PSO) to improve the performance of a well-known representation method of time series data which is the symbolic aggregate approximation (SAX), where PSO was used to propose a new weighted minimum distance WMD for SAX to recover some of the information loss resulting from the original minimum distance MINDIST on which SAX is based. WMD sets different weights to different segments of the time series according to their information content, where these weights are determined using PSO. We showed how SAX in conjunction with WMD can give better results in times series classification than the original SAX which uses MINDIST. In this paper we revisit this problem and propose optimizing WMD by using a hybrid of PSO and another bio-inspired optimization method which is Bacterial Foraging (BF); an effective bio-inspired optimization algorithm in solving difficult optimization problems. We show experimentally how by using this hybrid to set the weights of WMD we can obtain better classification results than those obtained when using PSO to set these weights.

**Keywords:** Bacterial Foraging, Particle Swarm Optimization, Symbolic Aggregate Approximation, Time Series Data Mining, Weighted Minimum Distance.

## 1 Introduction

A *time series* is an ordered collection of values measuring a certain phenomenon over a period of time. Formally, a time series  $S$  of length  $n$  is defined as;  $S = \langle s_1 = \langle v_1, t_1 \rangle, s_2 = \langle v_2, t_2 \rangle, \dots, s_n = \langle v_n, t_n \rangle \rangle$ . These values can be real numbers or multi-dimensional vectors.

Time series data arises in many applications including medical, financial, and engineering. For this reason, time series data mining has received attention over the last years.

Although time series data mining tackles several tasks such as query-by-content, clustering, classification, anomaly detection, motif discovery, and prediction, but

**Table 1.** The GEMINI algorithm for time series range queries

<p><b>Algorithm:</b> range_query(<math>Q, r</math>)</p> <ol style="list-style-type: none"> <li>1. Transform the time series in the database <math>D</math> from the original <math>n</math>-dimensional space into a lower dimensional space of <math>N</math> dimensions</li> <li>2. Define a lower bounding distance on the reduced space: <math>d^N(S_i, S_j) \leq d^n(S_i, S_j) \quad \forall S_i, S_j \in D</math></li> <li>3. Eliminate all the time series for which we have <math>d^N(Q, S) &gt; r \rightarrow</math> obtain a candidate response set</li> <li>4. Apply <math>d^n</math> to the candidate response set and eliminate all the time series that are farther than <math>r</math> from <math>Q</math> to get the true response set.</li> </ol>
--

executing all these tasks requires performing another fundamental task; *similarity search*. A similarity search problem consists of a database  $D$ , a query or pattern  $Q$ , which does not necessarily belong to  $D$ , and a constraint that determines the extent of proximity that the data objects should satisfy to qualify as answers to that query. Direct *sequential scanning* compares every single time series in  $D$  against  $Q$  to answer this query. Obviously this is not an efficient approach given that modern time series databases are usually very large.

The main frame to handle time series is through the *Generic Multimedia Indexing* (GEMINI) algorithm [4]. GEMINI was first presented for indexing time series. Later it was extended to other data types. GEMINI reduces the dimensionality of the time series by converting them from a point in an  $n$ -dimensional space into a point in an  $N$ -dimensional space, where  $N \ll n$ . A similarity measure is defined on the reduced space, which is a lower bound of the original similarity measure, thus the similarity search returns no false dismissals in this case. A post-processing sequential scan on the candidate response set is performed to filter out all false alarms and return the true response set. Table 1 illustrates the GEMINI algorithm.

*Dimensionality Reduction Techniques*, also known as *Representation Methods*, follow the GEMINI framework to find a faster solution to the similarity search problem in time series databases. This is achieved by projecting the original time series onto lower dimensional spaces, thus reducing their dimensionality, and then processing the query in those reduced spaces.

Several dimensionality reduction techniques have been suggested in the literature, of those we mention: *Piecewise Aggregate Approximation* (PAA) [7] and [20], *Piecewise Linear Approximation* (PLA) [12], and *Adaptive Piecewise Constant Approximation* (APCA) [8]. Other methods use multi-resolution approaches [11],

[15], [16], [17]; i.e. the time series are projected on several reduced spaces instead of a single reduced space.

One of the most important dimensionality reduction techniques of time series data is the *Symbolic Aggregate approXimation* method (SAX) [10]. This symbolic representation method uses pre-computed distances obtained from lookup tables. This makes SAX efficient and easy to apply.

In a previous work we used an optimization method- *Particle Swarm Optimization* (PSO) to propose a new minimum distance WMD for SAX to recover some of the information loss which results from using SAX as a dimensionality reduction technique. WMD sets different weights to different segments of the time series according to their information content. These weights are set through an optimization process. We have shown experimentally how WMD enhances the performance of SAX.

In this paper we revisit our previous work and optimize WMD by hybridizing PSO with another optimizer. This hybridization improves the performance of PSO in the aforementioned optimization problem.

The rest of the paper is organized as follows: in Section 2 we present related background. The new hybrid algorithms is introduced in Section 3 and evaluated in Section 4. In Section 5 we give concluding remarks.

## 2 Related Work

Piecewise Aggregate Approximation (PAA) was independently proposed in [7] and [20]. The basis of PAA is simple and straightforward, yet this method has been successfully used as a competitive method of dimensionality reduction technique of time series. PAA reduces the dimensionality of a time series  $S$  from  $n$  in the original space to  $N$  in the reduced space by segmenting the time series into equal-sized frames and representing each segment by the mean of the data points that lie within that frame.

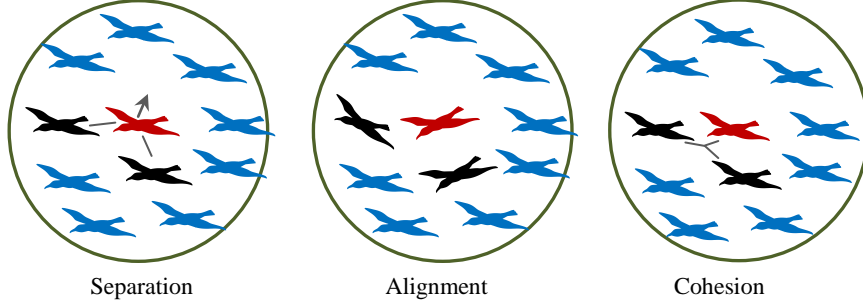
In [10] the authors proposed one of the most important symbolic representation methods of time series, which is the *Symbolic Aggregate approXimation* method (SAX). SAX assumes the Gaussianity of the normalized time series.

After normalizing the time series SAX projects the resulting time series on a lower-dimensional space using PAA as a dimensionality reduction technique, and then discretizes the outcome to obtain a sequence of symbols. These symbols are obtained by determining the number and locations of the breakpoints. Their locations are determined by using Gaussian lookup tables, whereas the user determines their number.

SAX uses the following similarity measure:

$$MINDIST(\hat{S}, \hat{R}) = \sqrt{\frac{n}{N}} \sqrt{\sum_{i=1}^N (dist(\hat{s}_i, \hat{r}_i))^2} \quad (1)$$

Where  $n$  is the length of the original time series,  $N$  is the length of the strings (the



**Fig. 1.** Simulating swarm's behavior

number of the segments),  $\hat{S}$  and  $\hat{R}$  are the symbolic representations of the two time series  $S$  and  $R$ , respectively, and where the function  $dist(\cdot)$  is implemented by using the appropriate lookup table.

In [13], we derived the *Weighted Minimum Distance* (WMD) from equation (1). WMD is defined as follows:

$$WMD(\hat{S}, \hat{R}) = \sqrt{\frac{n}{N}} \sqrt{\sum_{i=1}^N w_i (dist(\hat{s}_i, \hat{r}_i))^2}; \quad w_i \in [0,1] \quad (2)$$

The motivation behind this is that different segments have different information contents, so segments with a higher information content should be assigned a higher weight than those with a lower information content.

We showed in [13] that MINDIST is a special case of WMD, and we also showed that WMD is a lower bound of the Euclidean distance, so it guarantees no false dismissals.

The weights in (2) were obtained as the outcome of an optimization process where the optimizer we used was Particle Swarm Optimization (PSO). PSO is an evolutionary optimization algorithm inspired by the social behavior of some animals, such as bird flocking or fish schooling [6]. PSO is a member of a family of naturally-inspired optimization algorithms called *Swarm Intelligence* (SI) which are population-based optimization algorithms. In [19] a model is proposed to simulate a swarm. In this model individuals, also called *agents* or *particles*, follow three rules (Fig. 1); *Separation*: Each particle avoids getting too close to its neighbors. *Alignment*: Each particle steers towards the general heading of its neighbors, and *Cohesion*: Each particle moves towards the average position of its neighbors.

In PSO the particles are represented by vectors whose lengths are the dimension of the optimization problem at hand.

There are quite a large number of variations of PSO. In the following we present a standard PSO [5]. PSO starts by initializing a swarm of  $sSize$  particles at random positions  $\vec{X}_i^0$  and velocities  $\vec{V}_i^0$  where  $i \in \{1, \dots, sSize\}$ .

In the next step each position, and for each iteration, is evaluated using a *fitness function*, also called *objective function*.

The positions  $\vec{X}_i^{k+1}$  and velocities  $\vec{V}_i^{k+1}$  are updated at time step  $(k+1)$  according to the following formulae:

$$\vec{V}_i^{k+1} = \omega \vec{V}_i^k + \varphi_G (\vec{G}^k - \vec{X}_i^k) + \varphi_L (\vec{L}_i^k - \vec{X}_i^k) \quad (3)$$

$$\vec{X}_i^{k+1} = \vec{X}_i^k + \vec{V}_i^k \quad (4)$$

where  $\varphi_G = r_G \cdot a_G$ ,  $\varphi_L = r_L \cdot a_L$ ,  $r_G, r_L \rightarrow U(0,1)$ ,  $\omega, a_L, a_G \in \mathbb{R}$ ,  $\vec{L}_i^k$  is the best position found by particle  $i$ ,  $\vec{G}^k$  is the global best position found by the whole swarm,  $\omega$  is called the *inertia*,  $a_L$  is called the *local acceleration*, and  $a_G$  is called the *global acceleration*. These last three parameters are control parameters which are chosen by the algorithm designer. The algorithm continues until a stopping criterion terminates it.

As is the case with other evolutionary algorithms, PSO should keep a balance between *exploitation*, and *exploration*. Exploration is defined as the act of searching for the purpose of discovery, and exploitation is defined as the act of utilizing something for any purpose [1].

Diversity in PSO comes from two sources [21]; the first is the difference between the particle's current position and that of its best neighbor, and the other is the difference between the particle's current position and its best historical position.

Variation, although provides exploration, can only be sustained for a limited number of generations because convergence of the swarm to the best position is necessary to refine the solution (exploitation).

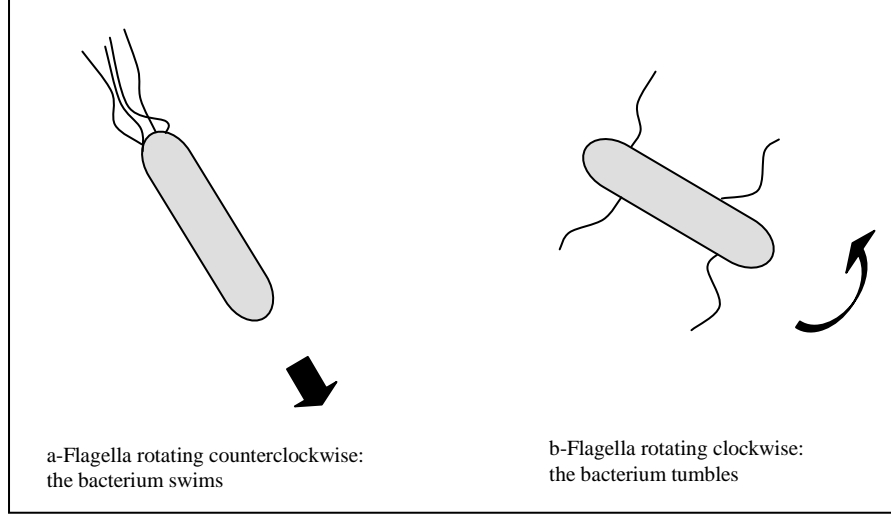
### 3 A Hybrid of Bacterial Foraging and Particle Swarm Optimization

In [2] the authors suggested hybridizing PSO with another member of the SI family, which is Bacterial Foraging. The new hybrid algorithm is called *Bacterial Swarm Optimization Algorithm* (BSOA). In their paper the authors tested BSOA on five well-known benchmark functions, and they showed that BSOA outperforms both BF and PSO. But before we introduce BSOA let us first present BF.

#### 3.1 Bacterial Foraging

Bacterial Foraging (BF) is a global optimization algorithm which has attracted increasing attention in the last few years. BF is another member of the SI family and it is inspired by the foraging behavior of the *Escherichia coli* (*E. coli*) bacteria.

The basis of BF is that natural selection tends to eliminate animals with poor foraging strategies and either replaces them with others that have better foraging strategies or shapes them into ones which have these desirable strategies [18]. BF formulates this process as an optimization problem.



**Fig. 2.** Chemotactic movements: (a) swimming (b) tumbling

In the presence of chemical attractants and repellants *E. coli* responds by two patterns of movement achieved by a set of flagella. When these flagella rotate in the clockwise direction the bacterium *tumbles*, whereas when they rotate in the counterclockwise direction the bacterium *swims*. These two movements are known as *chemotaxis*. Fig. 2 shows the tumbling and swimming chemotactic movements. The purpose of chemotaxis is to help the bacterium approach or avoid nutrient or noxious substance gradients. Sudden environmental changes may destroy the chemotactic progress causing the elimination and dispersal of a group of bacteria.

Given a function  $f(\theta); \theta \in \mathbb{R}^p$  ( $p$  is the number of parameters) to be minimized. BF finds the minimum of  $f$  by applying four mechanisms; chemotaxis, swarming, reproduction, and elimination-dispersal, which we will illustrate shortly, but let us first present this definition which is necessary to understand these mechanisms: a *chemotactic step* is a tumble followed by another tumble, or a tumble followed by a swim. Table 2 summarizes the symbols we are going to use to describe BF.

The position of each member of the population of  $N_b$  bacteria at the  $j^{th}$  chemotactic step,  $k^{th}$  reproduction step, and  $l^{th}$  elimination-dispersal event is denoted by

$$P(i, j, k) = \{\theta^i(j, k, l) / i = 1, 2, \dots, N_b\}.$$

We now describe the four mechanisms we mentioned earlier in this section:

- **Chemotaxis:** This is a key step in BF. Let  $\theta^i(j, k, l)$  be the  $i^{th}$  bacterium at the  $j^{th}$  chemotactic step,  $k^{th}$  reproduction step, and  $l^{th}$  elimination-dispersal event, then the movement of the bacterium can be represented by :

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \quad (5)$$

where  $\Delta$  is a vector in the random direction whose elements lie in the interval  $[-1, 1]$ .

- **Swarming:** *E. coli* demonstrate a swarming behavior in that they travel in rings of bacteria which move up the nutrient medium when they are placed in the center of a semisolid matrix with a single nutrient chemo-effector. When simulated by a high level of succinate the bacteria release an attractant aspartate which helps them aggregate into groups and thus move as a swarm. The cell-to-cell signal in the swam can be represented by the following function:

$$f_{cc}(\theta, P(j, k, l)) = \sum_{i=1}^{N_b} f_{cc}(\theta, \theta^i(j, k, l)) = \sum_{i=1}^{N_b} \left[ -d_{\text{attractant}} \cdot \exp\left(-\omega_{\text{attractant}} \sum_{m=1}^p (\theta_m - \theta_m^i)^2\right) \right] + \sum_{i=1}^{N_b} \left[ -h_{\text{repellant}} \cdot \exp\left(-\omega_{\text{repellant}} \sum_{m=1}^p (\theta_m - \theta_m^i)^2\right) \right] \quad (6)$$

where  $d_{\text{attractant}}$ ,  $\omega_{\text{attractant}}$ ,  $h_{\text{repellant}}$ ,  $\omega_{\text{repellant}}$  are coefficients to be chosen by the algorithm designer.

This objective function  $f_{cc}(\theta, P(j, k, l))$  is to be added to the original objective function to present a *time varying* objective function in that if many cells come close together there will be a high amount of attractant and hence an increasing likelihood that other cells will move towards the group. This produces the swarming effect [18].

- **Reproduction:** Through this process the least healthy bacteria die out and the healthier ones will undergo cell division to produce two daughter bacteria. This guarantees that the swam size will remain stable.
- **Elimination and dispersal:** There might be a gradual or sudden change in the environment where the bacteria live. As a result, the bacteria in a certain region are

**Table 2.** The symbols used in the description of bacterial foraging

$N_b$	The number of bacteria in the population
$N_c$	The number chemotactic steps
$N_s$	The swimming length
$N_{re}$	The number of reproduction steps
$N_{ed}$	The number of elimination-dispersal events
$P_{ed}$	The probability of elimination-dispersal
$C(i)$	The size of the step taken in the random direction determined by the tumble

killed or a group might be dispersed into another location. This has two effects on chemotaxis; the first is destroying the chemotactic progress. The second is that the new bacteria might be placed at locations with a better food source, thus assisting chemotaxis.

### 3.2 Bacterial Swarm Optimization Algorithm (BSOA)

The principle of BSOA is that local search is performed through the chemotactic movement operation of BFOA whereas the PSO operator executes the global search. This enables BSOA to keep a balance between exploration and exploitation [2].

In BSOA, and after undergoing a chemotactic step, the bacterium is mutated by a PSO operator which attracts the bacterium towards the global best position found so far by the whole population, and also towards its previous heading direction. Thus BSOA performs global search through its “PSO component” and local search through its “BF component”.

Our main motivation in using BSOA is that this hybrid optimization technique is particularly designed to handle a multi-modal optimization problem which is the case with the problem at hand as we showed in [14].

## 4 Experiments

The purpose of our experiments is to compare the performance of WMD in equation (2) when the weights are obtained by PSO (which is our work in [13]) with its performance when the weights are obtained by BSOA. We refer to the former as PSO\_WMD and to the latter as BSOA\_WMD.

As in [13], we conducted our experiments on a time series classification task on the first nearest-neighbor (1-NN) rule using leaving-one-out cross validation. The datasets on which we conducted our experiments are available at [9].

For each value of the alphabet size tested we first transform the time series to sequences by applying SAX (c.f. Section 2) and then we apply PSO\_WMD (or BSOA\_WMD). The weights  $w_i$  in relation (2) are obtained during a training phase; i.e. for each value of the alphabet size tested, we formulate an optimization problem whose fitness function is the classification error of the time series, and whose outcome is  $w_i$ .

In Table 3 we present some of the results we obtained for alphabet size equal to 3, 10, and 20, respectively.

As we can see from Table 3, the classification error of BSOA\_WMD is smaller than or equal to that of PSO\_WMD for all the datasets and for all values of the alphabet size tested, which shows that BSOA is indeed more effective in handling this optimization problem than PSO.



## 5 Conclusion

In this paper we revisited a previous work that applies particle swam optimization to set the weights of WMD – a weighted minimum distance applied in conjunction with SAX, and we used instead a hybrid optimization algorithm; bacterial swarm optimization algorithm – BSOA, to calculate these weights.

The experiments we conducted on a time series classification task show that BSOA\_WMD either outperforms or gives the same classification error obtained by applying PSO\_WMD on all the datasets and alphabet sizes tested. This shows that BSOA is effective in optimizing multi-modal problems.

**Table 3.** Comparison between PSO\_WMD and BSOA\_WMD

Dataset	Method	Alphabet Size	Classification Error
ECG	PSO_WMD	$\alpha=3$	0.19
		$\alpha=10$	0.08
		$\alpha=20$	0.10
	BSOA_WMD	$\alpha=3$	0.13
		$\alpha=10$	0.07
		$\alpha=20$	0.09
Trace	PSO_WMD	$\alpha=3$	0.51
		$\alpha=10$	0.41
		$\alpha=20$	0.30
	BSOA_WMD	$\alpha=3$	0.47
		$\alpha=10$	0.36
		$\alpha=20$	0.25
CBF	PSO_WMD	$\alpha=3$	0.36
		$\alpha=10$	0.07
		$\alpha=20$	0.06
	BSOA_WMD	$\alpha=3$	0.36
		$\alpha=10$	0.06
		$\alpha=20$	0.03
Beef	PSO_WMD	$\alpha=3$	0.73
		$\alpha=10$	0.53
		$\alpha=20$	0.43
	BSOA_WMD	$\alpha=3$	0.73
		$\alpha=10$	0.46
		$\alpha=20$	0.43
Lighting7	PSO_WMD	$\alpha=3$	0.68
		$\alpha=10$	0.36
		$\alpha=20$	0.37
	BSOA_WMD	$\alpha=3$	0.68
		$\alpha=10$	0.26
		$\alpha=20$	0.31
Fish	PSO_WMD	$\alpha=3$	0.88
		$\alpha=10$	0.51
		$\alpha=20$	0.22
	BSOA_WMD	$\alpha=3$	0.85
		$\alpha=10$	0.51
		$\alpha=20$	0.22

## References

1. Agnes, M., Webster's New World College Dictionary, Webster's New World, ISBN 0764571257, May 2004. (2004)
2. Biswas, A., Dasgupta, S., Das, S., Abraham, A.: Synergy of PSO and Bacterial Foraging Optimization-A Comparative Study on Numerical Benchmarks Innovations in Hybrid Intelligent Systems ASC, vol. 44, pp. 255–263. Springer, Heidelberg (2007)
3. Das, S., Biswas, A., Dasgupta, S., Abraham, A.: Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications. Foundations of Computational Intelligence 3, 23–55 (2009).
4. Faloutsos, C., Ranganathan, M., and Manolopoulos, Y.: Fast Subsequence Matching in Time-series Databases. In Proc. ACM SIGMOD Conf., Minneapolis (1994).
5. Fernández-Martínez, J.L. and García-Gonzalo, E.: What Makes Particle Swarm Optimization a Very Interesting and Powerful Algorithm? In B.K. Panigrahi, Y. Shi, and M.-H. Lim (Eds.): Handbook of Swarm Intelligence, ALO 8, pp. 37–65 (2011)
6. Haupt, R.L., Haupt, S. E.: Practical Genetic Algorithms with CD-ROM. Wiley-Interscience (2004)
7. Keogh, E., Chakrabarti, K., Pazzani, M. and Mehrotra: Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. J. of Know. and Inform. Sys. (2000)
8. Keogh, E., Chakrabarti, K., Pazzani, M., and Mehrotra, S. : Locally Adaptive Dimensionality Reduction for Similarity Search in Large Time Series Databases. SIGMOD pp 151-162 (2001)
9. Keogh, E., Zhu, Q., Hu, B., Hao, Y., Xi, X., Wei, L. & Ratanamahatana, The UCR Time Series Classification/Clustering Homepage: [www.cs.ucr.edu/~eamonn/time\\_series\\_data/](http://www.cs.ucr.edu/~eamonn/time_series_data/) C. A. (2011)
10. Lin, J., Keogh, E., Lonardi, S., Chiu, B. Y.: A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. DMKD 2003: 2-11(2003)
11. Megalooikonomou, C.: Multiresolution Symbolic Representation of Time Series. In proceedings of the 21st IEEE International Conference on Data Engineering (ICDE). Tokyo, Japan. (2005)
12. Morinaka, Y., Yoshikawa, M. , Amagasa, T., and Uemura, S.: The L-index: An Indexing Structure for Efficient Subsequence Matching in Time Sequence Databases. In Proc. 5th PacificAisa Conf. on Knowledge Discovery and Data Mining, pages 51-60 (2001)
13. Muhammad Fuad, M.M.: Particle Swarm Optimization of Information-Content Weighting of Symbolic Aggregate Approximation. In: Zhou, S., Zhang, S., Karypis, G. (eds.) ADMA 2012. LNCS (LNAI), vol. 7713, pp. 443–455. Springer, Heidelberg (2012)
14. Muhammad Fuad, M.M.: When Optimization Is Just an Illusion. In: Motoda, H., Wu, Z., Cao, L., Zaiane, O., Yao, M., Wang, W. (eds.) ADMA 2013, Part I. LNCS, vol. 8346, pp. 121–132. Springer, Heidelberg (2013)
15. Muhammad Fuad, M.M., Marteau, P.F.: Fast Retrieval of Time Series Using a Multiresolution Filter with Multiple Reduced Spaces. In: Cao, L., Feng, Y., Zhong, J. (eds.) ADMA 2010, Part I. LNCS, vol. 6440, pp. 137–148. Springer, Heidelberg (2010)
16. Muhammad Fuad, M.M., Marteau, P.F.: Multi-resolution Approach to Time Series Retrieval. Fourteenth International Database Engineering and Applications Symposium–IDEAS 2010 , 16-18 August, 2010, Montreal, QC, CANADA (2010)
17. Muhammad Fuad, M.M., Marteau, P.F.: Speeding-up the Similarity Search in Time Series Databases by Coupling Dimensionality Reduction Techniques with a Fast-and-dirty Filter.

Fourth IEEE International Conference on Semantic Computing– ICSC 2010, 22-24 September 2010, Carnegie Mellon University, Pittsburgh, PA, USA (2010)

18. Passino, K.M.: Biomimicry of Bacterial Foraging for Distributed Optimization and Control. IEEE Control Systems Magazine (2002).
19. Reynolds, C. W. : Flocks, Herds and Schools: A Distributed Behavioral Model. SIGGRAPH Comput. Graph. 21, 4 (1987)
20. Yi, B. K., and Faloutsos, C.: Fast Time Sequence Indexing for Arbitrary Lp Norms. Proceedings of the 26th International Conference on Very Large Databases, Cairo, Egypt (2000)
21. Zavala, A.M., Aguirre, A.H., and Diharce, E.V.: Robust PSO-based Constrained Optimization by Perturbing the Particle's Memory. Swarm Intelligence: Focus on ant and particle swarm optimization, Felix T. S. Chan and Manoj Kumar Tiwari, Ed. I-Tech Education and Publishing (2007)